

[First Hit](#) [Fwd Refs](#)
End of Result Set

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

☐ [Generate Collection](#) [Print](#)

L9: Entry 3 of 3

File: USPT

Oct 12, 1999

DOCUMENT-IDENTIFIER: US 5966714 A

TITLE: Method and apparatus for scaling large electronic mail databases for devices with limited storage

Abstract Text (1):

A method and an apparatus for scaling E-mail (electronic mail) address book databases for devices with limited storage capacity and synchronization of a first set of mail with a second set of mail at the message/folder level. A memory contains a mail subsetter for automatically generating a default personal address book and a mail synchronizer for application independent E-mail synchronization. A pseudo unique identification is generated for each message or folder in the mail boxes to be synchronized. An event log is then generated for each mail box. The memory also has a Synchronization mechanism for making the first set of data and the second set of data equivalent by using the information in the Change List generated by the Change Detection Mechanism. A processor runs the mail synchronizer.

Assistant Examiner (1):

Corrielus; Jean M.

Parent Case Text (1):

This application is a Continuation-In-Part of application Ser. No. 08/431,500 entitled APPLICATION INDEPENDENT RECORD LEVEL SYNCHRONIZATION filed Apr. 28, 1995, now U.S. Pat. No. 5,706,509.

Brief Summary Text (5):

E-mail (electronic mail) is one of the most popular data use application for data and voice telephones, in particular, for cellular telephones. However, telephones are typically limited in storage capacity. An average advanced cellular phone (such as a SmartPhone) being designed has less than one mega byte (MB) of storage available for use by a user. In contrast, a typical electronic address database using corporate environments are very large.

Brief Summary Text (8):

Additionally, MS Mail does not provide synchronization for the personal address book and the network database. For example, if John Doe is in both a user's personal address book and in a network database and if John Doe's name is deleted from the network database, his name will remain in the user's personal address book. Hence, the user will not know that John Doe is no longer available until a mail is transmitted to him and the mail is returned as "undeliverable" or "user unknown".

Brief Summary Text (9):

CC: Mail from Lotus Development Corporation provides synchronization but it is very inefficient. Synchronization is performed on-line with CC: Mail, i.e. when a user is connected and requires that the entire database be reconciled. Exemplary circumstances requiring reconciliation include when new records are added such as for new employees, when records have been deleted such as to account for employee attrition, and when updates are made such as for changed telephone numbers. Scanning a large database for change takes tens of minutes and since synchronization must be performed periodically, use of CC: Mail Remote or Mobile in a large organizational environment becomes impractical.

Brief Summary Text (10):

Therefore it is desirable to have a method and an apparatus for scaling E-mail address book databases for devices with limited storage capacity by providing an easy to use method for generating default personal address books automatically and to provide an efficient method for keeping the personal address books synchronized with a main database.

Brief Summary Text (12):

A method and an apparatus for scaling E-mail (electronic mail) address book databases for devices with limited storage capacity and synchronization of a first set of mail with a second set of mail at the message/folder level. A memory contains a mail subsetter for automatically generating a default personal address book and a mail synchronizer for application independent E-mail synchronization. A pseudo unique identification is generated for each message or folder in the mail boxes to be synchronized. An event log is then generated for each mail box. The memory also has a Synchronization mechanism for making the first set of data and the second set of data equivalent by using the information in the Change List generated by the Change Detection Mechanism. A processor runs the mail synchronizer.

Drawing Description Text (7):

FIG. 2a illustrates an exemplary block diagram of the mail synchronizer of the present invention.

Drawing Description Text (20):

FIG. 3i illustrates the general steps followed in processing changes in the mailbox (step 372 of FIG. 3h).

Drawing Description Text (21):

FIG. 3j illustrates the general steps followed in synchronizing PAB with the data source (step 373 of FIG. 3h).

Drawing Description Text (22):

FIG. 3k illustrates the general steps followed by the address synchronizer.

Drawing Description Text (27):

FIG. 5a is a block diagram illustrating an exemplary synchronization apparatus of the present invention.

Drawing Description Text (30):

FIGS. 5d-5f are alternate embodiments of a system block diagram illustrating the mail synchronization method and apparatus of the present invention.

Drawing Description Text (32):

FIG. 6a illustrates the general steps followed by the mail synchronizer of the present invention.

Drawing Description Text (33):

FIG. 6b describes the general steps followed by the present invention in generating a pseudo unique ID for each mail in a mail box to be synchronized with another mail box.

Drawing Description Text (40):

FIGS. 9a-9e, and 10a and 10b are flow charts describing the general steps followed by a Synchronization mechanism.

Drawing Description Text (42):

FIG. 12 is a table illustrating the mail synchronization method of the present invention.

Drawing Description Text (43):

FIG. 13 is a table illustrating exemplary mail synchronization scenarios where there are messages/folders with the same Key Field values.

Detailed Description Text (2):

The present invention provides a method and an apparatus for a storage efficient mechanism such that users can address E-mail off-line on their SmartPhones or Personal Digital Assistant's (PDA's), without devoting a large amount of storage space for a database or having to frequently consult a remote database for extra E-mail related information such as by looking up E-mail addresses on a network server. The method and the apparatus of the present invention involves automatically generating, storing and maintaining a subset of a large address book containing E-mail addresses locally on a client device with limited storage space such as a SmartPhone. The advantage of storing the E-mail addresses locally on the client device is that users can address and compose E-mail off-line. This reduces connection time and cost required to send messages. Periodically, the data in the personal address book is synchronized with data in a master address book. The present invention therefore provides an easy-to-use method for generating a default personal address book automatically and an efficient method for keeping a personal address book synchronized with a master database.

Detailed Description Text (4):

E-mail agent 100 runs continuously on a user's PC (personal computer) or network server 102, periodically checking for new messages. Additionally, E-mail agent 100 takes appropriate actions such as to notify the user if a message is urgent. E-mail agent 100 is coupled to MAPI (Messaging Application Programming Interface) 105 which is a standard API (Application Programming Interface) for transferring and manipulating mail messages, MS Mail 101 which is an application used to access mail through MAPI (Mail Application Program Interface), VIM (Vendor Independent Messaging) 109 which is also a standard API for transferring and manipulating mail messages and cc: Mail 103 which is an application based on VIM for accessing mail. The system is designed so that other mail API's and applications can be supported.

Detailed Description Text (5):

E-mail client application 104 runs on client device 106 such as a SmartPhone and allows a user to compose messages, access his or her network E-mails and to manage message folders. Periodically, E-mail client application 104 and E-mail agent 100 connect through a telephone call (or through serial cable, LAN, etc.) to synchronize their states.

Detailed Description Text (6):

FIG. 1b illustrates an exemplary system block diagram of the present invention. Computer 102 has memory 114 coupled to CPU 116. Memory 114 has mail synchronization 112 and various other elements 110 (also referred herein as subsetter 110) for subsetting a large E-mail address book to generate a smaller E-mail address book for client device 106. Further, computer 102 may have various peripheral devices 118 such as a keyboard and/or a display device. Computer 102 is also coupled to client device 106 as was illustrated in FIG. 1a.

Detailed Description Text (7):

FIG 1c illustrates various components of the present invention residing on host device 102 and client device 106 respectively. Host device (desktop or server) 102 has PIM (Personal Information Manager) 120 which stores contact names and numbers (i.e. addressing information). Host PC 102 also has creator 122 for initial creation of personal address book (PAB) 128; updater 123 for maintaining changes to PAB 128 based on new messages and changes in the network databases, mailbox 124 for storing all received and sent mail messages and MAB (master address book) 126 for storing E-mail directory information. MAB 126 may be a corporate address book if a user is on a LAN (local area network) or may be a personal "master address book" of a user if the user is at a stand alone PC. PAB (Personal Address Book) 128 in host device 102 is for storing addresses, names, numbers, etc., (i.e. addressing information). PAB 128 may be periodically updated for changes to its address entries. Source ID (identification) table 127 is a list of sources from which data for PAB 128 may be obtained. Sources of data for PAB 128 may be by user manual entry, from MAB 126, from PIM 120, etc. Each source is given a unique identifying number. Additionally, host device 102 also has DABI (Device Address Book Image) 130 which may be stored in PAB 128 and is equivalent to the size of the limited storage area of DAB (Device Address Book) 132 of client device 106.

Detailed Description Text (8):

More specifically, DABI 130 of host device 102 is a mirror image of DAB 132 of client device 106. In one embodiment of the present invention, DABI 130 is part of PAB 128 (i.e. a subset of

PAB 128). In this embodiment, whenever PAB 128 is updated, DABI 130 is automatically updated as well. In another embodiment of the present invention, DABI 130 may be separate from PAB 128. In this alternate embodiment, DABI 130 may be updated according to the changes made to PAB 128. In both embodiments, whenever DABI 130 of host device 102 is updated, DABI 130 of host device 102 and DAB 132 of client device 106 are synchronized with each other using address synchronizer 121. During synchronization, if an address in DAB 132 of host device 102 is no longer in DABI 130 of client device 106, then the corresponding entry in DAB 132 is deleted. Otherwise, if an address in DAB 132 of host device 102 is in DABI 130 of client device 106, then the entire corresponding entry in DAB 132 is replaced with the corresponding updated entry in DABI 130.

Detailed Description Text (9):

Client device 106 may be various remote client devices such as a SmartPhone, a PDA or a notebook PC. Both host PC 102 and client device 106 have an application (e.g. E-mail systems), app0 134 and appl 136 respectively, for which the subsetting and the synchronization of the address books are being performed.

Detailed Description Text (10):

PIM 120, Mailbox 124 and MAB 126 may reside on network server 127 instead of on host PC 102. More specifically, modules illustrated in host PC 102 may exist locally (on host PC 102) or remotely (on network server 127) and it is assumed that a network will take care of remote access if needed (i.e. for Mounting Network Volumes, Remote Procedure Calls, etc.). Such network implementation may be understood by a person skilled in the art given the detailed description of the present invention illustrated herein.

Detailed Description Text (14):

The names and mailing lists of cohorts communicated with are then used to match against the names in MAB 126 of a E-mail system such as CC: Mail 103 or MS Mail 101. The matching address entries are then automatically pulled out to form PAB 128. Each matching address entry 133 in PAB 128 contains the same information as the corresponding MAB 126 entry, such as phone numbers, mail stops, etc. Additional information may also be retrieved from other applications (i.e. sources) which include address books, such as external contact databases or personal information managers (PIMs) 120. For non-matching entries' (cohorts that are not in MAB 126) additional information such as phone number, mail stop, etc. for fields 135, may be retrieved from other sources (e.g. a contact database or PIM 120). The user may also be prompted to manually add additional information that was not included in MAB 126, the contact database, or PIM 120.

Detailed Description Text (17):

To keep PAB 128 synchronized automatically, the present invention stores a duplicate of the user's PAB 128 on his or her PC 102 or network server 127. When a new E-mail is received, the present invention automatically checks if the sender and other users copied on the message are in the compiled PAB 128. If so, the present invention updates the priority listing by updating address ranking score 139. If not, the addresses are added to PAB 128 in the same manner as described above. Periodically, such as once a day, the present invention checks the PC copy of the user's PAB 128 against MAB 126.

Detailed Description Text (20):

Occasionally, a user may need to address a person not in his or her PAB 128. In this case the user is given two options. The first option is to have E-mail client application 104 dial-up to connect to E-mail agent 100 and browse the entries stored in PAB 128, MAB 126, or PIM 120. The second option is to allow the user to manually type and use an address without searching any address books. The latter option is useful for new contacts that would not be in an address book.

Detailed Description Text (24):

The additional fields selected in ECCO contain home phone numbers, cellular phone numbers and mail addresses. As entries are found in the ECCO database, they are also copied into PAB 128. Along with each file stored in PAB 128, there is a corresponding identifier 137 that indicates the source of the information contained in that field (MAB 126, PIM, Database, Manual Entry, etc.). The identifier is used later for synchronization of changes back to the source as will

be described later. After scanning all known sources of additional information, there may still be empty fields. The user can optionally enter this information manually.

Detailed Description Text (26):

FIGS. 2a through 2d, and 5 through 13 illustrate an exemplary method and apparatus for data synchronization which may be adapted for use in the present invention's address synchronization process. Given the detailed description disclosed herein, such implementation of the address synchronization process of the present invention may be readily understood by a person skilled in the art.

Detailed Description Text (27):

FIG. 2a is a block diagram illustrating a mail synchronizer. Mail synchronizer 112 synchronizes changes made independently on an e-mail system on two separate computers or two different e-mail systems on the same computer. Pseudo unique ID (identification) generator 200 generates a pseudo unique ID for each mail or folder in a mail box for an e-mail system. Event log generator 202 generates an event log by generating events for each ID (i.e. for each mail or folder). Synch mechanism 204 synchronizes the mail boxes using the generated event logs.

Detailed Description Text (29):

FIG. 2c is a block diagram of the event log generator illustrated in FIG. 2a. Event log generator 202 has mail event log generator 210 and folder event log generator 212. Mail event log generator 210 generates an event for the event log for each mail in mail boxes to be synchronized. Folder event log generator 212 generates an event for the event log for each folder in the mail boxes to be synchronized.

Detailed Description Text (30):

FIG. 2d is a block diagram of the synch mechanism of the present invention illustrated in FIG. 2a. Synch mechanism 204 has sub-parts, Change Detection mechanism 230, Error Detection mechanism 234, Change Existing Data mechanism 238 and Synchronization mechanism 236. Change Detection mechanism 230 detects the changes which have occurred to a given mail box since the last synchronization. Error Detection 234 identifies incorrect updates and creates performed by Change Detection mechanism 230. Error conditions triggering Error Detection 234 include when a record's Key Attribute has changed in MB0' and/or when more than one message/folder is sharing the same Key Attribute values (an example of a non-unique identifier case is described in more detail below). Synchronization mechanism 236, performs the synchronization of data sets, given information regarding the mail boxes produced by Change Detection mechanism 230. Change Existing Data 238 modifies existing data to make the mail boxes equivalent.

Detailed Description Text (34):

FIG. 3c illustrates the general steps followed to create PAB 128 (step 302 illustrated in flow diagram FIG. 3a). In step 331, a mailbox is opened. In step 332, it is determined whether or not another folder exists. If another folder does not exist, then in step 333, the mailbox is closed. In step 334, PAB 128 is sorted by ranking. If there is another folder, then in step 335, the mail folder is processed. In step 336, the folder is opened. In step 337, it is determined whether there is another message. If there is no other message, then in step 339, the folder is closed. Otherwise if there is another message, then in step 338, the message is processed and the general steps are repeated.

Detailed Description Text (41):

FIG. 3g illustrates general steps in creating DAB 132 (step 304 of flow diagram of FIG. 3a). In step 364, the address synchronizer mechanism 121 is allowed to create DAB 132 using DABI 130. The addresses and additional information that has been modified in DABI 132 are then recorded in a change list as additions, deletions or modifications. As this is the initial creation of DAB 132, the entire change list will contain only additions.

Detailed Description Text (42):

Address synchronizer mechanism 112 (also referred herein as mail synchronizer 112) processes change requests by creating a change list (see FIG. 4b for change list example 392). The change list consists of a list of E-mail addresses, commands, time stamp, and data. The E-mail address is used to uniquely identify the entry that will be changed. The command is used to identify if

this is a deletion, addition, or modification. The time stamp is used to resolve synchronization conflicts. This occurs if changes are made on both the device and the desktop without synchronization between the changes. For example, if in the contact database someone changes Frank's mail address, and on the SmartPhone the user changes his phone number, both changes are processed. First, a modification that indicates the mail address is adjusted and then a modification that indicates that the phone number is adjusted are made. The end results is as though both changes occurred. However, it should be noted that if two changes are made to the same field, only the latter of the changes will be of effect.

Detailed Description Text (43):

The changes in the change list may be processed whenever a connection is made. Therefore if a user creates a dial-up connection to browse a large E-mail message, synchronization of DABI 132 with DAB 130 can occur. This synchronization can also be performed over serial or LAN connection. The user can therefore make adjustments on the phone and the changes to DAB 132 will be synchronized with PAB 128. More detail on synchronization can be gained by the descriptions accompanying FIGS. 5 through 13 and by referring to pending application 08/431,500 APPLICATION INDEPENDENT RECORD LEVEL SYNCHRONIZATION by one of the co-inventors of the present invention, Michael Man-Hak Tso, and assigned to the assignee of the present invention. Said application is hereby incorporated by reference.

Detailed Description Text (44):

FIG. 3h describes the general steps followed by the update mechanism (updater 123 illustrated in FIG. 1c) of the present invention. In step 371, PAB 128 is synchronized with the MAB, PIM, contact database and other data sources indicated by the source ID table. A more detailed description of this step is illustrated in the flow diagram of FIG. 3j. In step 372, the changes in the mailbox is processed. A more detailed description of this step is illustrated in the flow diagram of FIG. 3i. In step 373, if changes occur in the PAB 128, then the re-synchronization is performed between the PAB 128, DABI 132 and DAB 130 as described earlier (i.e. a more detailed description of this step is illustrated in the flow diagram of FIG. 3j).

Detailed Description Text (45):

FIG. 3i illustrates the general steps followed in processing changes in the mailbox (step 372 of FIG. 3h). In step 385, new messages in the mailbox is detected. In step 386, the message is processed. The message is processed following the general steps which are illustrated for the flow diagram and the accompanying text for FIG. 3d and new addresses are added. In step 387, if there are changes in PAB 128, then PAB 128 is updated accordingly. In step 388, if there are changes in PAB 128, then DABI 132 is updated and the change is propagated to the DAB 130 through the use of the address synchronizer 112.

Detailed Description Text (46):

FIG. 3j illustrates the general steps followed in synchronizing PAB 128 with a data source (where data source can be the source of data such as a PIM, a MAB, or manual entry by a user, etc.). An exemplary source ID table 391 illustrating exemplary source IDs and the corresponding data sources is illustrated in FIG. 4a. The general steps illustrated here are a continuation from step 373 of FIG. 3h. In step 374, the next entry from PAB 128 is retrieved. In step 375, the source ID for the address field of the entry retrieved is obtained. The PAB example 390 of FIG. 1d illustrates exemplary fields which may be associated with each entry in PAB 128 and the corresponding source ID. In step 376, the data source identified by the source ID retrieved is checked for the address.

Detailed Description Text (47):

In step 377, if the address no longer exists in the data source, then in step 378, the entry is deleted from PAB 128. (i.e. entry may no longer be valid because the user associated with the particular address is for example no longer with the company associated with the data source or the user has a new address, etc.) Otherwise, if the address is in the data source, then in step 379, the data source ID for each of the additional information fields are retrieved. (i.e. if the address listed in the PAB for that entry is also listed in the corresponding data source, then the user associated with the address is assumed to still be with the company/organization associated with the data source and it is concluded that the address is valid.) In step 380, the address used to retrieve the source ID is used to access entries in the source. The entries

in the data sources are checked for changes to the data (i.e. if there are discrepancies between the data in the fields of PAB 128 and the corresponding data in the data source). In step 381, if there are changes to the data in the data sources, then updates are made to PAB 128 or the data source. In step 382, the choice of changing the PAB entry or the data source is made by using a modification time stamp on the PAB entry. If PAB 139 has been modified since the last update with data source, then the data source is updated similar to the PAB entry. Back in step 374, the next entry from PAB 128 is retrieved in order to synchronize PAB 128 entries with the data source. As described earlier, in one implementation of the present invention, DABI 130 is a subset of PAB 128 (i.e. part of PAB 128), in which case updating of PAB 128 is equivalent to updating DABI 130. DAB 132 on a client device such as SmartPhone 102 is then synchronized with DABI 130 using address synchronizer 121. In an alternate embodiment, DABI 130 is separate from PAB 128. In this case, once PAB 128 is updated using the data sources as described above, DABI 130 is updated using PAB 128.

Detailed Description Text (48):

FIG. 3k illustrates the general steps followed by the address synchronizer. In step 391, changes to DAB and DABI are queued up until a connection is made between the two. In step 392, when a connection is made (and user settings indicate synchronization should be performed), then the change lists are processed. In step 393, the two change lists are merged and processed in order (the oldest changes are processed first).

Detailed Description Text (49):

FIG. 4a illustrates an exemplary source ID table 410. FIG. 4b illustrates an exemplary change list 420 for DAB 132. FIG. 4c illustrates an exemplary change list for DABI 130. FIG. 4d illustrates an exemplary merged list created by merging and sorting by time the change list for DAB illustrated in FIG. 4b and the change list for DABI illustrated in FIG. 4c. Further details on how the information between DABI 130 and DAB 132 are synchronized are described in the remainder of the specification.

Detailed Description Text (50):

FIG. 5a illustrates an exemplary mail synch mechanism (also referred herein as an address synchronizer) apparatus. The apparatus has Change Detection mechanism 230 with inputs of mail boxes (or an address book) MB0', MB1' and MB0 or MB1. Change Detection mechanism 230 produces Change List CL0 for MB0 and MB0' and Change List CL1 for MB1 and MB1'. A Change List (CL) is a list of address/message/folder changes for a given mail box address book or (described in more detail in the description accompanying FIG. 5c below). Synchronization mechanism 236 is fed inputs MB0', MB1' and newly produced Change Lists CL0 and CL1. The output of Synchronization mechanism 236 is input to Change Existing Data 238 to produce synchronized mail boxes MB0" and MB1".

Detailed Description Text (51):

FIG. 5b is a flow diagram illustrating the general steps followed by the mail synch mechanism apparatus of the present invention referenced in FIG. 5a. In step 241, address books or mail boxes MB0, MB1, MB0' and MB1' are input to Change Detection mechanism 230. Address books or mail boxes MB0 and MB1 are equivalent and may be the same file or non existent. MB0' and MB1' are a version of MB0 and MB1 after they have been independently modified. They are saved when synchronization was last run (i.e. when the last time steps 241 through 245 were performed). Change Detection mechanism 230 determines the changes which have been made to address books or mail boxes MB0' and MB0 between synchronizations to produce Change List CL0. In step 242, Change Detection mechanism 230 performs the same operation to produce Change List CL1 from address books or mail boxes MB1' and MB1. In step 243, Synchronization mechanism 236 takes as input, CL0, CL1, MB0' and MB1'. The result of Synchronization mechanism 236 is input to Change Existing Data 238 to produce synchronized address books or mail boxes MB0" and MB1". In step 244, original address books or mail boxes MB0 and MB1 are deleted. In step 245, MB0" and/or MB1" are saved as MB0 and MB1 for the next synchronization process.

Detailed Description Text (54):

Finally, column 253 indicates whether the synchronization has taken place. The entries for this column initially indicates "no" for corresponding address/messages/folders which have not been processed by Synchronization mechanism 236 (see FIGS. 9a-9e for the general steps followed by

Synchronization mechanism 236). After processing by Synchronization mechanism 236, the column entries are changed to "yes" to indicate that the corresponding address/message/folder has been synchronized.

Detailed Description Text (57):

Although a combination of the operations, Created, Updated and Deleted, may be applied to a record, only the end result at the time of the next synchronization is relevant for the Change List. This may be illustrated with an exemplary address book or mail box MB0'. Address book or mail box MB0' contains the final form of each address/message/folder before the next synchronization and is a mail box containing all the changes made to an original mail box MB0. If an address/message/folder Ru was created in MB0', then modified, the correct change appearing on the Change List CL0 for MB0 and MB0' is Created since the original version of the address book or mail box MB0 did not contain Ru. Thus, even if Ru was modified after creation in address book or mail box MB0', for the purposes of listing the changes made between the original address book or mail box MB0 and the modified version MB0', the difference represented in the list is that a new message/folder Ru was created.

Detailed Description Text (59):

FIGS. 5d-5f are exemplary embodiments of a system block diagram with the implementation of the address book synchronization. The present invention may be used to synchronize addresses/messages/folders between address books or mail boxes MB0 and MB1, belonging to application app0 and application app1 respectively. A variety of configurations are possible. For example, MB0 may reside in a satellite device (e.g. a notebook or a hand held computer, such as an Apple.RTM. Newton, a Sharp.RTM. Wizard, or a Casio.RTM. BOSS) and MB1 may reside on a host computer (e.g. a desktop or a notebook PC) as illustrated in FIG. 5d. Further, MB0 and MB1 may reside on the same system as illustrated in FIG. 5e. MB0 and MB1 may also reside on two different PC's linked by a computer network as illustrated in FIG. 5f. In addition, app0 and app1 may be the same application. The present invention may be implemented for synchronization of any two or more address books or mail boxes and is not limited to the exemplary configurations illustrated herein.

Detailed Description Text (60):

More specifically, FIG. 5d illustrates an embodiment of the present invention where neither app0 nor app1 generates a Change List (CL). Host PC 500 is coupled to satellite device 504 via cable 502. Synchronizer 112 may reside on either host PC 500 or satellite device 504 or on both

Detailed Description Text (61):

In an alternative embodiment, one of the applications, app0 or app1, may support synchronization by generating a Change List (CL).

Detailed Description Text (62):

FIG. 5e illustrates an embodiment where synchronizer 112 is implemented as an application residing on PC 500 separate from the applications/e-mail systems being synchronized. This allows users to synchronize data between existing installed applications without having to buy new versions of the applications. Further, users do not have to wait for application vendors to add synchronization support to their applications. The only requirement for this embodiment is that the applications being synchronized provide a means for a third party apparatus to import and export data to and from their native data formats.

Detailed Description Text (63):

Examples of such means include but are not limited to: a published file format, an application programming interface (API), a specialized interface such as Standard Query Language (SQL) used by databases, a Dynamic Data Exchange (DDE), or some communication protocol for transferring information to and from a mobile or remote device. The details on how to implement the importation and exportation of data to and from an application using published formats are application specific and well understood by those skilled in the art.

Detailed Description Text (64):

In an alternative embodiment illustrated in FIG. 5f, synchronizer 112 is implemented as part of

one of the applications residing on PC 500 and involved in the synchronization. The only requirement here is that appl (residing on PC 506) with which app0 (residing on PC 500) is to be synchronized must provide a means for a third party apparatus to import and export data to and from their native formats.

Detailed Description Text (65):

In yet another alternate embodiment, synchronizer 112 may be implemented as part of both app0 and appl, and a predefined communication protocol is used to exchange information such as for a Change List. Other embodiments are possible and are included in the scope of this invention. For these alternative embodiments, the apparatus and methods of the present invention are unchanged, and the only difference is in how the apparatus is packaged. Given the present invention, how the apparatus may be packaged will be readily understood by a person skilled in the art.

Detailed Description Text (66):

Since the synchronization method of this invention is only concerned with the end resulting difference between MB0' and MB0, rather than the history of all changes that might have happened, a synchronization aware application (such as Appl in FIG. 5f) need not keep a traditional log of all activities for each record. Instead, it is safe to truncate the log such that only the most recent relevant changes are kept. The state transition diagrams in FIGS. 5g and 5h illustrate how the log or more specifically, a Change List as referred to herein, can be truncated such that there is exactly one change flag marked for each record. Truncating the Change List is desirable because it reduces the amount of storage required for the Change List. This is particularly important for memory limited devices, such as hand held computers.

Detailed Description Text (67):

In FIG. 5g, after synchronization at state 530, all addresses/messages/folders are marked "No Change address/." A "No Change address/" message/folder may transition to either "Updated address/" or "Deleted." In state 532, an "Updated" message/folder stays "Updated" unless it becomes "Deleted". In state 534, a "Deleted" address/message/folder always remains "Deleted."

Detailed Description Text (68):

In FIG. 5h, a newly created address/message/folder is marked "Created" as illustrated in state 540 and remains "Created" unless it is later deleted. If a "Created" address/message/folder is deleted, it is marked "Created.sub.-- Deleted" in state 542. Even if a new address/message/folder is later modified, it is still marked "Created" for synchronization purposes. Created.sub.-- Deleted address/messages/folders are transient and are always ignored during synchronization.

Detailed Description Text (69):

FIG. 6a illustrates the general steps followed by the e-mail synchronizer of the present invention. In step 601, for each address or mail, a pseudo ID is generated. More detailed description and steps followed by this portion is described in FIG. 6b in the accompanying text. In step 602, once a pseudo ID is generated for each address or mail, an event log is generated per each address book or mail box. A more detailed description is provided in FIGS. 6c, 6d, 6e and 6f in the accompanying text. Once event logs are generated, they are synchronized in step 603. A more detailed description is provided in FIGS. 9a through 9e.

Detailed Description Text (70):

FIG. 6b describes the general steps followed by the present invention in generating a pseudo unique ID for each address or mail in an address book or mail box to be synchronized with another address book or mail box. In step 611, an attribute in each mail which is most likely to be unique is selected. An example of such an attribute is the mail message reception time. In step 612, an ID is generated from the selected attributes for each address or mail. For example, in a typical or mail system such as for CC: Mail, a message has the following attributes: message type, folder name, sender, recipient, subject, receiving time, priority, receipt request, and size. The receiving time may be selected first for generating a pseudo unique ID. The attributes may be selected in the following order if the receiving time is not unique: sender, subject, size, recipient, folder name, message type, priority and receipt request.

Detailed Description Text (72):

Although the above process does not guarantee to generate a unique ID for each address/message/folder, the process will not destroy the synchronization process. If a user copies an address or message to the same folder, this will produce two identical addresses or messages. Because all their attributes are the same, the two address or messages will have the same Id. For any operation such as delete, update or new (copy) operation performed on any one of these identical addresses or messages, the operation may be applied to any other address or message having the same ID at the synchronization site. Because the operations are performed to addresses or messages having the same identical attributes, the result is the same. After a delete or an update operation, every address or message becomes unique because the duplicated addresses or messages are deleted by the delete operation and the update operation causes identical addresses or messages (which have not been deleted) to have different attribute values through the updated attributes.

Detailed Description Text (77):

FIGS. 7a and 7b are flow diagrams describing the general steps followed by the Change Detection mechanism referenced in FIG. 2d. Change Detection mechanism 230 deduces all the changes that have occurred in a given address book or mail box since the last synchronization. This is accomplished by first saving a copy of the address book or mail box at the end of a synchronization. At the next synchronization, the addresses/messages/folders are compared in the modified address book or mail box (e.g. MB0') with the saved address book or mail box (e.g. MB0), detecting the changes which must have happened since the last synchronization. In the case where the synchronization is run for the first time, there are no addresses/messages/folders in the saved address book or mail box (e.g. MB0), and Change Detection mechanism 230 concludes that all current addresses/messages/folders have been created.

Detailed Description Text (86):

In error case 1, the correct result to be produced by Change Detection mechanism 230 is that Rn' is Updated. Since the Key Attribute is changed during the synchronization process, assuming the changed Key Attributes are still unique, Change Detection mechanism 230 deduces that Rn was Deleted, and Rn' was Created. A Delete and a Create is equivalent to an Update. Since Rn' also contains all the unmodified attributes in Rn, as well as any attributes a user or an application normally chooses not to synchronize. Thus no correction is needed. The case where Rn's Key Attributes are the same as one or more other record's Key Attributes is handled as an error case 2 described below.

Detailed Description Text (87):

For error case 2, Change Detection mechanism 230 deduces incorrect Updates and Creates. For example, assuming that addresses/messages/folders R1 and R2 have the same Key Attribute values and that R1' and R2' are unchanged, i.e. R1' has all the same attributes as R1 and R2' as R2. If Change Detection mechanism 230 is run on R2' before R1', it is possible for pseudo unique id generator 200 to erroneously determine that R1 has been updated to R2' and that R2 has been updated to R1'. The correct determination which should be made by pseudo unique id generator 200 is that neither R1 nor R2 has changed. Similarly, if R1' is unchanged but R2' has changed, it is possible for Change Detection mechanism 230 to determine that R1 has been updated to R2' and R2 has been updated to R1'. In either case, after results are produced by Change Detection mechanism 230, if R1 or R2 contains attributes not normally synchronized, R1's unsynchronized attributes would be exchanged with R2's unsynchronized attributes.

Detailed Description Text (96):

FIGS. 9a through 9e are flow charts describing the general steps followed by the Synchronization mechanism referred to in FIG. 2d. Addresses/Messages/Folders Rn and Rm are exemplary addresses/messages/folders in address book or mail box MB0 and addresses/messages/folders Rx and Ry are exemplary addresses/messages/folders in mail box MB1.

Detailed Description Text (97):

Error Correction mechanisms one and two generate a list of changes when comparing a changed mail box MB0' to a saved address book or mail box MB0. When synchronizing address books or mail

boxes MB0' and MB1', Error Correction two, first processes address books or mail boxes MB0' and MB0, producing Change List 0 (CL0), and then processes address books or mail boxes MB1' and MB1, producing Change List 1 (CL1). Address books or mail boxes MB0 and MB1 may or may not be data from the same application or have the same format. However, address books or mail boxes MB0 and MB1 are equivalent (i.e. synchronized) since they are saved at the end of the last synchronization and there may be only one address book or mail box saved.

Detailed Description Text (107):

In FIG. 9e, for each address/message/folder Rx in CL1, if Rx is not marked as DONE in step 975, then in step 976 it is determined if Rx is marked as CREATE. If Rx is marked as CREATE, then in step 977, Rx is added to MB0' (using the steps illustrated in FIG. 11a). In step 981, Rx is marked as DONE. Back in step 976, if Rx is not marked as CREATE, then in step 978 it is determined if Rx is marked as UPDATE. If Rx is marked as UPDATE, then in step 979 Rx is updated in MB0' (using the steps illustrated in FIG. 11a). In step 981, Rx is marked as DONE. Back in step 978, if Rx is not marked as UPDATE, and if Rx is marked as DELETE in step 980, then Rx is deleted in MB0' (using the steps illustrated in FIG. 11a). In step 981, Rx is marked as DONE. The steps from step 974 to step 980 are repeated until all addresses/messages/folders Rx in CL1 are processed. At this point, the addresses/messages/folders in MB0' and MB1' are equivalent, i.e. synchronized.

Detailed Description Text (108):

With the availability of a complete history of the changed data as produced above, it is possible to make automatic decisions for conflict resolution. In addition, there is little danger of incorrectly overwriting or deleting data. This allows synchronization to run unobtrusively in the background without having to ask a user to confirm before overwriting or deleting data.

Detailed Description Text (109):

FIGS. 11a through 11c are flow diagrams describing the general steps followed by the Change Existing Data mechanism referred to in FIG. 2d. Although the details for importing and exporting data is application specific and well understood, making the required changes to the address book or mail boxes without unique address/message/folder identifiers is non-trivial. Such case is resolved by Change Existing Data mechanism 238.

Detailed Description Text (115):

FIG. 12 is a table 440 illustrating the Synchronization mechanism 236 of the present invention. Column 442 of table 440 indicates case numbers. Column 444 indicates the changes in address book or mail box MB0. Column 446 indicates the changes in address book or mail box MB1. Column 448 indicates the result of the changes in address book or mail box MB0. Column 450 indicates the result of the changes in address book or mail box MB1. R1+ indicates that R1 has been added (i.e. created) to the data set. R1- indicates that R1 has been deleted from the data set. R1->R1' indicates that R1 has been changed (i.e. updated) to R1'.

Detailed Description Text (116):

Change Detection mechanism 230 produces the cases as described in column 444 for changes to address book or mail box MB0 and column 446 for changes to address book or mail box MB1. Columns 448 and 450 describe what Synchronization mechanism 236 must perform given CL0 and CL1. More specifically, column 448 describes the action resulting in the changes to address book or mail box MB0 and column 450 describes the action resulting in the changes to address book or mail box MB1. If the additional inputs to Synchronization mechanism 236 are MB0 and MB1, then all actions described in columns 448 and 450 must be performed. If the additional inputs are MB0' and MB1', then only those actions which are circled must be performed. It is assumed for the sake of illustration that all addresses/messages/folders R1, R2, R3, etc., have unique Key Attribute values.

Detailed Description Text (118):

What has been described is a method and an apparatus for a storage efficient mechanism such that users can address E-mail off-line on their SmartPhones, without devoting a large amount of storage space for a database or having to frequently consult a remote database for extra E-mail related information such as by looking up E-mail addresses on a network server. The method and

the apparatus of the present invention involves storing a subset of a large address book containing E-mail addresses locally on a SmartPhone (herein referred to as a personal address book). The advantage of storing the E-mail addresses locally on the SmartPhone is that users can address and compose E-mail off-line. This reduces connection time and cost required to send messages. Periodically, the data in the personal address book is synchronized with data in a master address book. The present invention therefore provides an easy-to-use method for generating a default personal address book automatically and an efficient method for keeping a personal address book synchronized with a master database.

CLAIMS:

1. An apparatus for scaling large electronic or mail databases for devices with limited storage, said apparatus comprising:

a storage device which contains,

a mail subsetter configured to automatically generate a personal address book containing a first set of address entries, said first set of address entries obtained from at least one data source, and

a mail synchronizer configured to synchronize said first set of address entries with a second set of address entries, said mail synchronizer configured to produce the same synchronization results regardless of whether one or both of said first and second sets of address entries are modified prior to synchronization, said mail synchronizer coupled to said mail subsetter; and

a processor for running said mail subsetter and said mail synchronizer coupled to said processor.

2. The apparatus of claim 1 wherein said mail subsetter comprises:

a device address book image containing said first set of addresses;

a creation mechanism configured to create said personal address book and said device address book image and coupled to said personal address book; and

an update mechanism configured to update said personal address book from a data source using said mail synchronizer and by processing new mail and coupled to said personal address book.

7. The apparatus of claim 1 wherein said mail synchronizer comprises:

a pseudo unique identification generator configured to generate an identification for each said message/folder;

an event log generator configured to generate an event for an event log for each said mail/folder with said identification, said event log generator coupled to said pseudo unique identification generator; and

a synchronization mechanism configured to synchronize said first set of mail and said second set of mail to be equivalent, said synchronization mechanism coupled to said event log.

8. The apparatus of claim 1 wherein said synchronization mechanism comprises a change detection mechanism configured to generate a change list for said first and second sets of mail, said change list listing the changes made at a message/folder level to said first and second sets of data.

11. An apparatus for scaling large electronic mail databases for devices with limited storage, said apparatus comprising:

storage means which contains,

means for automatically generating a personal address book containing a first set of address entries, said set address entries obtained from at least one data source, and

means for synchronizing said first set address entries and a second set of address entries equivalent by using the information in said change list generated by said change detection mechanism, said means for synchronizing producing the same synchronization results regardless of whether one both of said first and second sets of address entries are modified prior to synchronization, said means for synchronizing coupled to said means for generating; and

means for running said means for automatically generating and said means for synchronizing, said means for running coupled to said storage means.

12. The apparatus of claim 11 wherein said means for generating comprises of:

means for containing said first set of address entries;

means for creating said personal address book and said means for containing said first set of address entries and coupled to said personal address book; and

means for updating said personal address book from said at least one data source using said means for synchronizing and by processing new mail and coupled to said personal address book.

19. A system for scaling large electronic mail databases for devices with limited storage, said system comprising:

a storage device which contains,

mail subsetter configured to automatically generate a default personal address book containing a first set of address entries, said first set of address entries obtained from at least one data source, and

a mail synchronizer configured to synchronize said first set of address entries with a second set of address entries; and

a processor for running said mail subsetter and said mail synchronizer, said processor coupled to said storage device.

20. The system of claim 19 wherein said mail subsetter comprises of:

a device address book image containing said first set of addresses;

a creation mechanism configured to create said personal address book and said device address book image and coupled to said personal address book; and

an update mechanism for updating said personal address book from a data source using said mail synchronizer and by processing new mail and coupled to said personal address book.

25. The system of claim 19 wherein said mail synchronizer comprises:

a pseudo unique identification generator configured to generate an identification for each said message/folder;

an event log generator configured to generate an event for an event log for each said mail/folder with said identification, said event log generator coupled to said pseudo unique identification generator; and

a synchronization mechanism configured to synchronize said first set of mail and said second set of mail to be equivalent, said synchronization mechanism coupled to said event log.

26. The system of claim 19 wherein said synchronization mechanism comprises a change detection

mechanism configured to generate a change list for said first and second sets of mail, said change list listing the changes made at a message/folder level to said first and second sets of data.

29. A method for scaling large electronic mail databases for a client system with limited storage, said method comprising the steps of:

automatically generating a personal address book containing a first set of address entries, said first set of address entries obtained from at least one data source; and

synchronizing said first set of address entries with a second set of address entries using a first change list generated for said first set of address entries and a second change list generated for said second set of address entries, said step of synchronizing producing the same synchronization results regardless of whether one or both of said first and second sets of address entries are modified prior to synchronization.

30. The method of claim 29 wherein said step for generating comprises of:

creating said personal address book and said first set of address entries; and

updating said personal address book from said at least one data source using said step for synchronizing and by processing new mail.

38. The method of claim 37 further comprising the step of opening a mailbox to retrieve new message.

47. The method of claim 46 further comprising the step of synchronizing said device address book and said device address book image with said personal address book every time said personal address book is changed.

48. A system for scaling large electronic mail databases for devices with limited storage, including code stored on a computer-readable medium and executable by a computer, comprising:

a mail subsetter code configured to automatically generate a personal address book containing a first set of address entries, said first set of address entries obtained from at least one data source, and

a mail synchronizer code configured to synchronize said first set of address entries with a second set of address entries, said mail synchronizer configured to produce the same synchronization results regardless of whether one or both of said first and second set of address entries are modified prior to synchronization, said mail synchronizer code coupled to said mail subsetter code.

50. The method of claim 29 wherein said step of synchronizing further comprises the step of generating a first change list comprising a list of changes made to said first set of address entries and a second change list comprising a list of changes made to said second set of address entries.

52. The method of claim 29 wherein said step of synchronizing further comprising the step of changing messages/folders in said first set of mail and messages/folders in said second set of mail for synchronizing messages/folders in said first set of mail and said second set of mail.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)